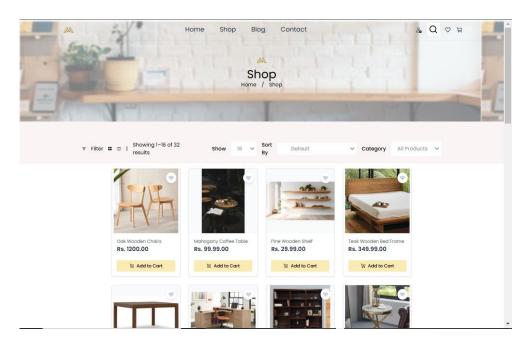
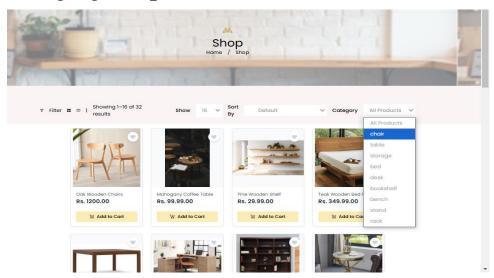
# Day 4 - Dynamic Frontend Components-Furniro

## ScreenShot of:

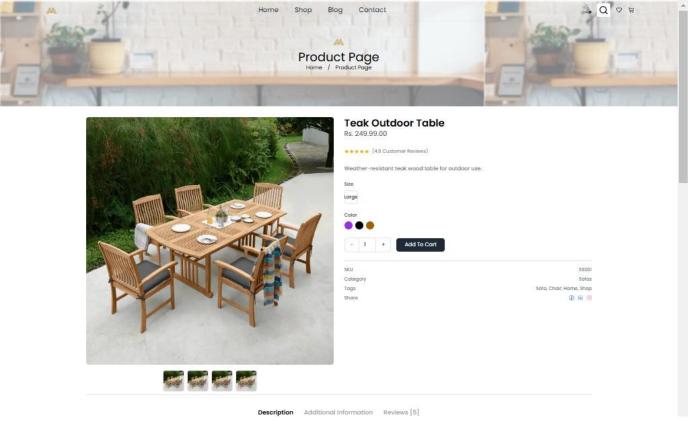
\*Product Listing Component:



# \* Category Component:



# \*Product Detail Component:

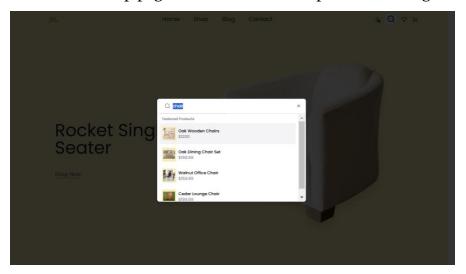


## \*Search Bar:

## This Screenshot is of the Nav bar .



This is from Shop page where user can find products through Search Bar

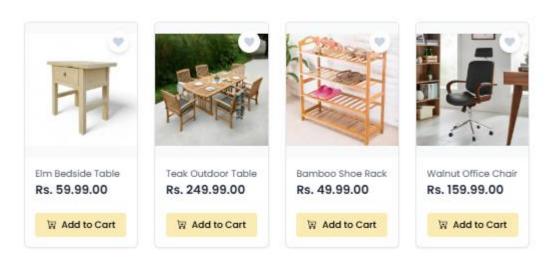


#### \*Similar Products:

. Find this similar products by same tags through API.

# **Top Picks for You**

find a bright ideal to suit your taste with our great selection of suspension, floor and table lights



- A search bar is implemented using the SearchCommand component.
- The handleSearch function filters products based on the search query. It checks the name field of each product and show them.
- The search bar dynamically updates the filteredData as the user types.

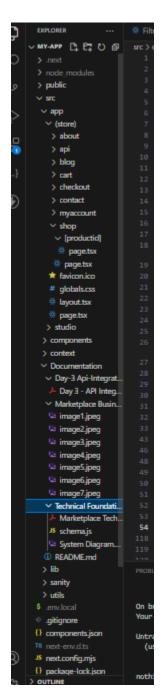
#### Code:

```
| Description |
```

## **Product Details (Dynamic Routes):**

- Shows a product image, name, description, price and a discount percentage for real-time.
- Includes a "View Product" button linking to the product's detailed page using its \_id.

#### **File Structure:**



#### Code:

```
EXPLORER
                --- S Filter.tsx
                                  T8 FetchProduct.ts × ⊕ page.tsx
> public
 ∨ src
                     Codeium: Refactor | Explain | Generate JSDoc | X

sport const getAllProducts = async () -> {
  ∨ app

✓ (store)
                      Execute Query

const ALL_PRODUCTS_QUERY - defineQuery(\*[_type -- "product"]{
    > about
                              _id,
    > api
                              name,
    > blog
                              description,
    > cart
                             price,
    > checkout
                              tags,
                              sizes,
    > contact
                              "imageUrl": image.asset->url,
    > myaccount
                              rating,
    > shop
                             discountpercentage,
    ★ favicon.ico
                             category,
isFeaturedProduct,
    # globals.css
    stockLevel.

    page.tsx

   > studio
  > components
                                const products = await sanityFetch({ query: ALL_PRODUCTS_QUERY });
  > context
  > Documentation
                               return products.data || [];
                            catch (error) {
  console.error("error fetching all products " + error);
  return [];
  > lib
   > пь

∨ queries

 18 FetchInventory.ts
    18 FetchProduct.ts
                      31 export const getProductById = async (id: string): Promise<Product | null> => {
    18 FetchSales.ts
  name,
description,
price,
  > utils
                              tags.
 gitignore
                            tags,
sizes,
"imageOrl": image.asset->url,
rating,
discountpercentage,
 () components.json
 JS next.config.mjs
                              category.
 () package json
                     PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
 JS postcss.config.mjs

    README.md

                      On branch main
 18 sanity.cli.ts
                     Your branch is up to date with 'origin/main'.
 18 sanity.config.ts
 T8 sanity.types.ts
                     Untracked files:
```

### **Product List:**

- Fetches data from a Sanity CMS backend for items of type "food".
- Displays each product in a card format, showing its name, description, price, discount, and an image.

### **Dynamic Routing**

Dynamic routing is used to handle routes like /shop/:id for individual product pages.

#### 1. Create Dynamic Route File:

```
In the pages directory, create a [id].tsx file inside the shop folder: pages/shop/[id].tsx
```

#### 2. Get Product Data Dynamically:

## Final CheckList

Frontend Component Development	Styling and Responsiveness	Code Quality	Documentation and Submission	Final Review
<b>√</b>	<b>√</b>	✓	✓	<b>\</b>